

HW7

黄佳溢

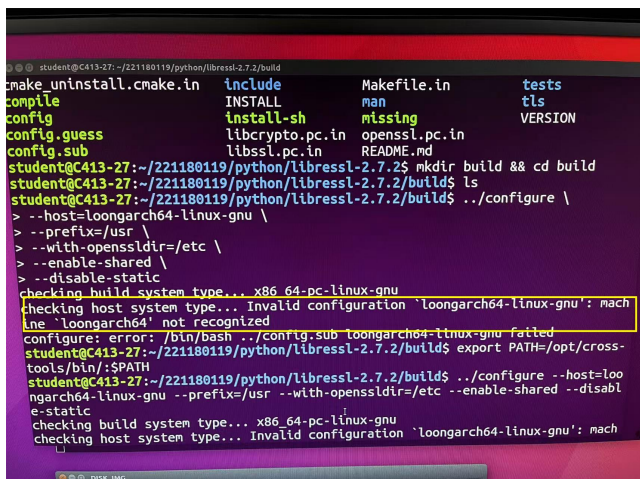
December 24, 2024

1 Python生成及搬运

这个部分在课件上也已经说明详尽了，只需要依样画葫芦即可，这里仅点出需要注意的几个事项；

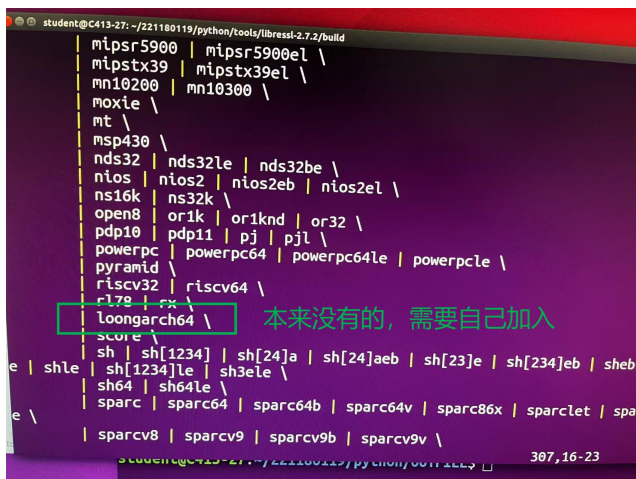
1. 加入龙芯支持项

我们在执行configure配置时总会出现”**mechine 'loongarch64' not recognized**”，这是因为工具库源码并不支持龙芯架构，我们只需要为其添加支持权限即可。具体操作为寻找源码中的config.sub文件，输入”/powerpc”可以跳转到支持架构的位置，在那里加入”loongarch64”即可。



```
student@C413-27: ~/221180119/python/libressl-2.7.2/build
make uninstall.cmake.in include Makefile.in tests
compile INSTALL man
config install-sh missing VERSION
config.guess libcrypto.pc.in openssl.pc.in
config.sub libssl.pc.in README.md
student@C413-27:~/221180119/python/libressl-2.7.2$ mkdir build && cd build
student@C413-27:~/221180119/python/libressl-2.7.2/build$ ls
student@C413-27:~/221180119/python/libressl-2.7.2/build$ ./configure \
> --host=loongarch64-linux-gnu \
> --prefix=/usr \
> --with-openssldir=/etc \
> --enable-shared \
> --disable-static
checking build system type... x86_64-pc-linux-gnu
checking host system type... Invalid configuration 'loongarch64-linux-gnu': machine 'loongarch64' not recognized
configure: error: /bin/bash ./config.sub loongarch64-linux-gnu failed
student@C413-27:~/221180119/python/libressl-2.7.2/build$ export PATH=/opt/cross-tools/bin/:$PATH
student@C413-27:~/221180119/python/libressl-2.7.2/build$ ./configure --host=loongarch64-linux-gnu --prefix=/usr --with-openssldir=/etc --enable-shared --disable-static
checking build system type... x86_64-pc-linux-gnu
checking host system type... Invalid configuration 'loongarch64-linux-gnu': machine 'loongarch64' not recognized
```

Figure 1: 报错



```
student@C413-27: ~/221180119/python/tools/libressl-2.7.2/build
mipsr5900 | mipsr5900el \
mipstx39 | mipstx39el \
mn10200 | mn10300 \
moxie \
mt \
msp430 \
nds32 | nds32le | nds32be \
nios | nios2 | nios2eb | nios2el \
ns16k | ns32k \
open8 | or1k | or1knd | or32 \
pdp10 | pdp11 | pj | pjl \
powerpc | powerpc64 | powerpc64le | powerpcle \
pyramid \
riscv32 | riscv64 \
r178 | rx \
loongarch64 \
score \
sh | sh[1234] | sh[24]a | sh[24]aeb | sh[23]e | sh[234]eb | sheb
shle | sh[1234]le | sh3ele \
sh64 | sh64le \
sparc | sparc64 | sparc64b | sparc64v | sparc86x | sparclet | sparcv8 | sparcv9 | sparcv9b | sparcv9v \
307,16-23
```

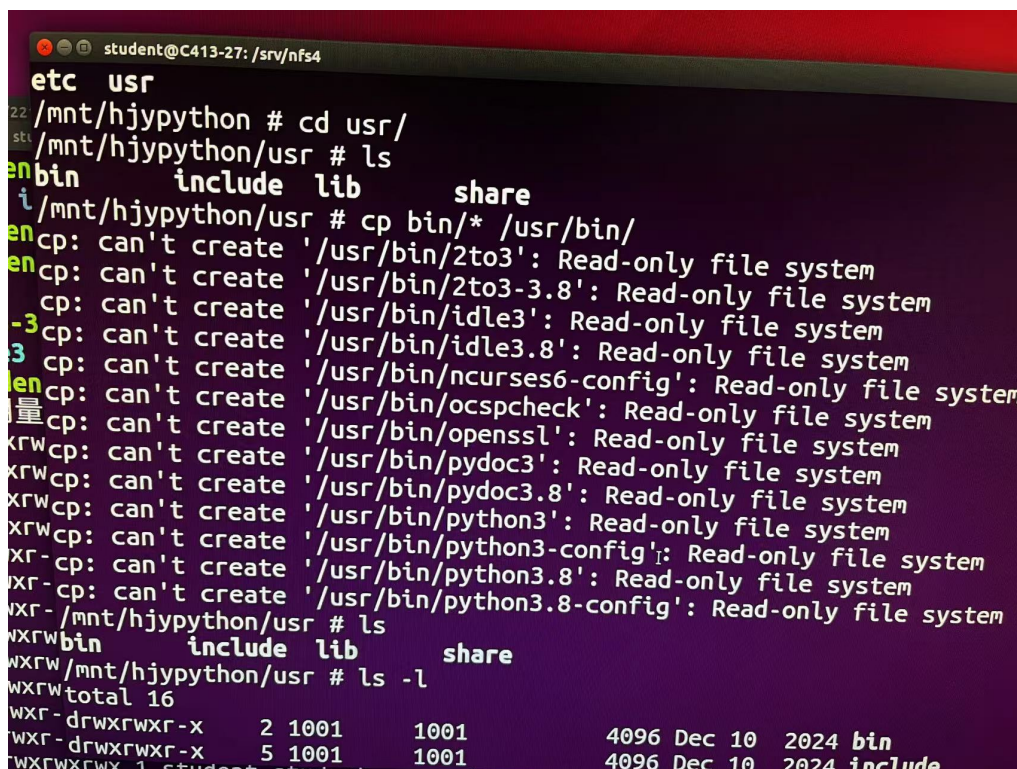
Figure 2: 修改

还需要注意的是，① 为了方便寻找config.sub的位置，我们可以调用find -name config.sub指令来加快效率； ② 在config.sub文件中，有时候输入powerpc也不一定能直接

跳转到具体的位置，此时可以按n键来跳转到下一个位置[1]；

2. 无法搬运

这种情况常常出现在，在嵌入式开发板中，将/target/usr/bin目录下的东西复制到/usr/bin目录下的过程，具体报错见图



```
student@C413-27: /srv/nfs4
etc usr
/mnt/hjpython # cd usr/
/mnt/hjpython/usr # ls
bin include lib share
/mnt/hjpython/usr # cp bin/* /usr/bin/
cp: can't create '/usr/bin/2to3': Read-only file system
cp: can't create '/usr/bin/2to3-3.8': Read-only file system
cp: can't create '/usr/bin/idle3': Read-only file system
cp: can't create '/usr/bin/idle3.8': Read-only file system
cp: can't create '/usr/bin/ncurses6-config': Read-only file system
cp: can't create '/usr/bin/ocspcheck': Read-only file system
cp: can't create '/usr/bin/openssl': Read-only file system
cp: can't create '/usr/bin/pydoc3': Read-only file system
cp: can't create '/usr/bin/pydoc3.8': Read-only file system
cp: can't create '/usr/bin/python3': Read-only file system
cp: can't create '/usr/bin/python3-config': Read-only file system
cp: can't create '/usr/bin/python3.8': Read-only file system
cp: can't create '/usr/bin/python3.8-config': Read-only file system
/mnt/hjpython/usr # ls
bin include lib share
/mnt/hjpython/usr # ls -l
total 16
-rwxr-xr-x 2 1001 1001 4096 Dec 10 2024 bin
-rwxr-xr-x 5 1001 1001 4096 Dec 10 2024 include
```

Figure 3: Read-only file system

这是由于我们在BootLoader的交互式启动环节，启动的代码为：

```
g console=ttyS0, 115200 root=/dev/ram0
```

这表明嵌入式系统将默认为仅写入的权限，所以我们就无法对其进行读写操作 解决方法就是在启动的时候，在后面加入**rw**。

```
g console=ttyS0, 115200 root=/dev/ram0 rw
```

3. 没内存

```
student@C413-27: /srv/tftpboot
/mnt # ls
app          driver1.ko  hjpython
/mnt # cd hjpython/
/mnt/hjpython # ls
etc usr
/mnt/hjpython # ls
etc usr
/mnt/hjpython # cd usr/
/mnt/hjpython/usr # ls
bin include lib share
/mnt/hjpython/usr # cd bin/
/mnt/hjpython/usr/bin # ls
2to3          ncurses6-config  pydoc3.8          python3.8-config
2to3-3.8      ocspcheck        python3           python3-config
idle3         openssl          python3-config    python3.8
idle3.8       pydoc3           python3.8
/mnt/hjpython/usr/bin # cd ..
/mnt/hjpython/usr # cp bin/* /usr/bin/
cp: write error: No space left on device
cp: write error: No space left on device
cp: write error: No space left on device
cp: write error: No space left on device
/mnt/hjpython/usr #
```

Figure 4: No space left on device

这个问题因人而异，有些人在配置文件系统时，默认的给定的设备空间比较大，所以不会出现这个问题（感谢前辈）。我比较倒霉，遇到的设备空间比较小QAQ，这是一个很难绷的问题，首先我需要解释这个现象出现的原因：

① 在嵌入式开发板中，除了挂载的mnt外，其他的文件目录都是占用了开发板自己的硬盘空间，在文件系统配置比较小的时候，加上提前创建好的其他文件，剩余空间已经岌岌可危，亲测剩下的空间仅能装入几个.c文件，但是对于python这种大可执行文件而言，是完全无法转移的。

② 挂载成功之后的mnt目录比较特殊，它的底层是NFS（网络文件系统），是我们的主机提供给开发板的共享文件夹，所使用的是主机的空间，所以我们在开发板终端下，在mnt目录做的一切操作都是在主机空间里进行，所以不会出现空间不足的问题。

为解决这个问题，这里我给出两种思路：①（正规做法）重装文件系统，将分配空间做大；②（非正规做法）将所有环境变量都指向mnt目录，在mnt目录下操作。这里我选择**第二种方案**（嘻嘻）并在下一个章节为你解释这种做法的可行性。


```
student@C413-27: /mnt/python_loongarch/lib
令 'cd5' 来自于包 'cd5' (universe)
libffi.so.7.0.0  libnss_files.so  rcrt1.o

/mnt/python_loongarch/bin # echo $LD_LIBRARY_PATH
/mnt/python_loongarch/bin # export PATH=/mnt/python_loongarch/lib:$PATH
/mnt/python_loongarch/bin # ./python3.8
/python3.8: error while loading shared libraries: libcrypt.so.1: cannot open shared object file: No such file or directory
/mnt/python_loongarch/bin # export LD_LIBRARY_PATH=/mnt/python_loongarch/lib:$LD_LIBRARY_PATH
/mnt/python_loongarch/bin # echo $LD_LIBRARY_PATH
PATH
/mnt/python_loongarch/bin # echo $LD_LIBRARY_PATH
/mnt/python_loongarch/lib:
/mnt/python_loongarch/bin # ./python3.8
/python3.8: error while loading shared libraries: libpthread.so.0: cannot open shared object file: No such file or direy
/mnt/python_loongarch/bin # ./python3.8
/python3.8: error while loading shared libraries: libdl.so.2: cannot open shared object file: No such file or directory
/mnt/python_loongarch/bin # ./python3.8
/python3.8: error while loading shared libraries: libutil.so.1: cannot open shared object file: No such file or directoy
/mnt/python_loongarch/bin # ./python3.8
/python3.8: error while loading shared libraries: libutil.so.1: cannot open shared object file: No such file or directoy
Python 3.8.2 (default, Dec 17 2024, 16:57:47)
[GCC 8.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> prubt
```

Figure 6: 调试

2.3 实现效果

下面展示了交互界面运行和.py文件执行的效果

```
student@C413-27: /mnt/python_loongarch/lib
令 'cd5' 来自于包 'cd5' (universe)
libffi.so.7.0.0  libnss_files.s

/mnt/python_loongarch/bin # python3.8
python 3.8.2 (default, Dec 17 2024, 16:57:47)
[GCC 8.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print("hello world")
hello world
>>> exit()
/mnt/python_loongarch/bin #
```

Figure 7: 交互界面运行效果

```
/mnt # ls
hello.py          python_loongarch
/mnt # python3 hello.py
hello world!
My name is MI!
/mnt #
```

Figure 8: 运行.py文件效果

3 总结

本学期的数字系统II实验到这里也就圆满结束了，在本课程中，我深入学习了嵌入式系统的运行原理，为我将来的科研打下了坚实的基础（撒花）在这里还是感谢所有为我提供帮助的老师，助教和同学！

唯一可惜的是，由于实验的文件体积较大（达到了6.3GB）所以无法将我的全部代码进行开源QAQ，但是我还是会将所有的实验报告上传到Github仓库！为南京大学的开源社区贡献一份力量！（耶）